REMARKS

The following remarks are deemed fully responsive to the office action mailed December 22, 2004. Claim 1 is modified to correct a grammatical error. No new matter is added. Claims 1 – 20 remain pending, of which claims 1, 11, 19 and 20 are independent.

Claim Rejections – 35 U.S.C. § 102

The examiner makes the following references to U.S. Patent Number 5,999,736 granted to Gupta et al. (hereinafter "Gupta"): col. 26 lines 20-43; col. 26 line 20 – col. 27 line 50; col. 26 line 44 – col. 27 line 50; fig. 12; col. 24 line 5 – col. 25 line 24; col. 26 lines 25-53; col. 24 line 42 – col. 25 line 24; col. 27 lines 11-30; and col. 24 lines 5-34. Gupta, however, only includes figures 1-11 and columns 1-22. We are therefore *unable to follow or comprehend the Examiner's arguments with respect to Gupta*. Clarification or withdrawal of the Gupta reference is requested.

Nevertheless, we have reviewed Gupta and point out the following differences between Gupta and the immediate application.

The immediate application teaches a processor with two or more parallel instruction paths that process instructions; the instruction paths may be implemented as a programming core within an EPIC processor, and on a common die. See paragraph [0019]. FIG. 1 of the immediate application shows architecture 10 with an array of execution paths 12(1-N) that process instructions through execution units 14(1-N), respectively. Each of the instruction paths 12 has an array of pipelined execution units 14. Two or more parallel instruction paths process the same program thread with different optimization characteristics. Assessment logic monitors the processing of the initial program thread and selects the heuristics defining which path is in the lead. The lead path continues processing the initial thread without being disturbed and the other paths are reallocated, or synchronized, with the optimization characteristics of the lead instruction path. See paragraph [0007] of the immediate specification.

On the other hand, Gupta discloses a method and apparatus for optimizing execution of code by first executing the code to generate path profiling information,

5

then relocating a plurality of instructions to another location, and then re-executing the code. See Gupta abstract and summary. Gupta asserts that "two types of profiling are relevant to an embodiment of the present invention", and that "'Edge profiling' tracks the number of times each edge in the program flow graph is traversed while 'path profiling' tracks the number of times various acyclic paths in a program are traversed." See Gupta column 4, lines 58-62. Gupta clearly requires that code is first executed to generate profiling information. The code is then optimized by **moving** instructions from one location to another, and then the program is **re-executed**. Gupta thereby pertains to code optimization, as opposed to processor optimization as taught by the immediate application. Gupta is therefore non-analogous to performance optimization of multi-core processors.

Gupta specifically recites that "FIG. 2 illustrates a typical computer system 200 in which the present invention operates." See Gupta col. 3, lines 14-15. Gupta also recites that "processor 202 may be any of a wide variety of general purpose processors or microprocessors such as a Pentium® processor manufactured by Intel® Corporation." See Gupta col. 3, lines 38 – 40. It is therefore apparent that the system of Gupta is not a processor; nor does it specifically concern any particular processor design. Moreover, the processors cited by Gupta are not multi-core and therefore do not have multiple instruction pipelines. In fact, Gupta does not disclose, anywhere, the use of parallel pipelines.

Referring now to the inventions of the immediate application, claim 1 recites a method for optimizing the processing of instructions through a processor, including the steps of:

a) processing first like instructions through two or more instructions paths of the processor, each of the paths having different heuristics associated therewith;

b) monitoring progress of the first like instructions through the instruction paths;

c) determining which of the instruction paths is a first leader in processing the first like instructions; and

d) modifying heuristics of one or more of the instruction paths based on heuristics of the first leader.

Clearly, Gupta does not disclose a method that optimizes the processing of instructions through a processor. Gupta does not, for example, disclose or suggest (a)

6

processing of instructions through instruction paths, (b) determining which of the instruction paths is a first leader, or (c) modifying heuristics of one or more of the instruction paths. Gupta cannot, therefore, anticipate claim 1.

Claim 2 recites grouping the first like instructions as a bundle from a common program thread. Instructions are bundled within instruction cache 22 and are then issued by an instruction issue section 26. See paragraph [0021] of the specification. Gupta does not disclose or suggest instruction bundling as required by claim 2.

Claim 3 recites modifying heuristics of each of the instruction paths. Claim 4 recites modifying heuristics of instruction paths other than heuristics of the leader. Gupta does not disclose or suggest modifying heuristics of instruction paths within the processor.

Claim 5 recites processing the first like instructions through the leader without being affected by the step of modifying. Claim 6 recites processing additional instructions from a program thread of the first like instructions through the multiple instruction paths and without redundancy. Claim 7 recites processing second like instructions through two or more instructions paths of the processor, each of the paths having different heuristics associated therewith, monitoring progress of the second like instructions through the instruction paths, determining which of the instruction paths is a second leader in processing the instructions, and modifying heuristics of one or more of the instruction paths based on heuristics of the second leader. As argued above, Gupta does not disclose or suggest multiple instruction paths or modifying heuristics thereof.

Claim 8 recites modifying one or more of CPU-bound heuristics and memory-bound heuristics. Claim 9 recites modifying heuristics based upon one or more of branch prediction and prefetch heurisitcs. Claim 10 recites asymptotically approaching optimized characteristics for the instruction paths. Nowhere does Gupta modify one or more of CPU-bound heuristics and memory-bound heuristics, modify heuristics based upon one or more of branch prediction and prefetch heuristics or asymptotically approaching optimized characteristics for the instruction paths.

Gupta simply does not disclose or suggest these features of claims 1-10. We respectfully requests reconsideration and allowance of claims 1-10.

Claim Rejections – 35 U.S.C. § 103

Claims 11 – 20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent Number 6,715,062 granted to Moore (hereinafter "Moore") in view of Gupta. Respectfully we disagree.

When applying 35 U.S.C. §103, the following tenets of patent law must be adhered to:

    a) The claimed invention must be considered as a whole;

    b) The references must be considered as a whole and must suggest the desirability and thus the obviousness of making the combination;

    c) The references must be viewed without the benefit of impermissible hindsight vision afforded by the claimed invention; and

    d) Reasonable expectation of success is the standard with which obviousness is determined. MPEP §2141.01, *Hodosh v. Block Drug Co., Inc.*, 786 F.2d 1136, 1134 n.5, 229 U.S.P.Q. 182, 187 n.5 (Fed. Cir. 1986).

In addition, it is respectfully noted that to substantiate a *prima facie* case of obviousness the initial burden rests with the Examiner who must fulfill three requirements. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the references or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on Applicant's disclosure. (emphasis and formatting added) MPEP § 2143, *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

We again note that the Examiner's references to Gupta are, by and large, non-existing; and thus again ask for clarification or withdrawal of the Gupta reference.

Nonetheless, we present the following arguments against Moore in view of Gupta based upon our reading and understanding of Gupta.

Claim 11 recites a processor for processing program instructions, including:

a) at least two parallel instruction paths, each of the paths having an array of pipeline execution units and associated heuristics affecting how the instructions are processed therein; and

b) assessment logic for monitoring processing of the instructions within the paths and for modifying the heuristics of at least one of the paths to improve per thread performance of the processor.

Moore discloses a processor and method for performing a hardware test during instruction execution in a normal mode. See Moore title. In particular, FIG. 1 of Moore shows a processor 10 with one instruction sequencing logic 13; logic 13 includes instruction fetch address register (IFAR) 30 that contains an effective address (EA) indicating a cache line of instructions to be fetched and processed from L1 I-cache 18. See Moore col. 2, lines 49 – 52. Thus, Moore's processor 10 has only one instruction path.

Moore, therefore, does not disclose "at least two parallel instruction paths" where each path has an array of execution units, as required by claim 11. Further, Moore does not show execution units with associated heuristics that affect how instructions are processed therein, as also required by claim 11. By way of illustration, the immediate application discloses that each instruction path 12 may be implemented as a programming core within an EPIC processor, on a common die. See paragraph [0019] of the immediate application.

On the other hand, Moore discloses a processor whereby no-op instructions, which may have existed or have been inserted into the instruction stream, are replaced by test instructions that test the processor. Moore does not therefore disclose execution units with heuristics that affect how instructions are executed.

As argued above, Gupta does not disclose or suggest modifying heuristics of execution units to improve thread performance. Gupta relocates a plurality of instructions to another location, and then re-executes the code. See Gupta abstract and summary. Thus, Gupta discloses modifying code and does not disclose modifying heuristics of an execution unit of a processor. Gupta's code optimization and Moore's replacement of no-op instructions with test instructions do not, when combined, render claim 11 (or anything similar thereto). Gupta and Moore cannot therefore

9

render claim 11 obvious under 35 U.S.C. §103. Reconsideration of claim 11 is respectfully requested.

Claims 12 – 18 depend from claim 11 and benefit from like arguments. However, these claims have additional features that patentably distinguish over Moore and Gupta. For example, claim 12 recites the heuristics of each of the instruction paths having one or more of fetch heuristics, execution heuristics, and cache heuristics. Moore and/or Gupta do not mention fetch heuristics, execution heuristics and cache heuristics. Moore and Gupta cannot, therefore render claim 12 obvious.

Claim 13 recites the two parallel instruction paths comprising parallel core processors on a common die. As argued above, neither Moore nor Gupta disclose parallel instruction paths.

Claim 14 recites the parallel instruction paths constructed and arranged to initially process first like instructions therethrough, the assessment logic monitoring the processing of the first like instructions to determine optimized heuristics for the instruction paths. Neither Moore not Gupta disclose assessment logic monitoring the processing of the first like instructions to determine optimized heuristics for the instruction paths.

Claim 15 recites the parallel instruction paths constructed and arranged to subsequently process different instructions therethrough to improve per thread processing performance. Again, Moore and/or Gupta do not disclose parallel instruction paths.

Claim 16 recites that the parallel instruction paths are constructed and arranged to subsequently process second like instructions therethrough, the assessment logic monitoring the processing of the second like instructions to determine optimized heuristics for the instruction paths. Again, neither Moore nor Gupta disclose parallel instruction paths, assessment logic or determining optimized heuristics for the instruction paths.

Claim 17 recites each of the parallel instruction paths forming a cluster constructed and arranged to process instructions as bundles. Neither Moore nor Gupta disclose parallel instruction paths, grouping the parallel instruction paths into clusters or processing instructions as bundles.

10

Claim 18 recites parallel instruction paths and assessment logic cooperating to process one or more bundles of like instructions through the instruction paths to monitor and then modify heuristics of the instruction paths to improve per thread processing of the instructions. As argued above, neither Moore nor Gupta disclose modifying heuristics of the parallel instruction paths.

In sum, Moore and Gupta, alone or in combination, do not disclose the features of claims 12-18.; they cannot therefore render claims 12-18 obvious under 35 U.S.C. § 103. Reconsideration of claims 12 – 18 is respectfully requested.

Claim 19 recites a processor of the type having at least two parallel instruction paths, each of the paths having an array of pipeline execution units and associated heuristics affecting how the instructions are processed, the improvement including assessment logic for monitoring processing of the instructions within the paths and for modifying the heuristics of at least one of the paths to improve per thread performance of the processor. As argued above, Moore and Gupta do not disclose modifying heuristics of one or more of at least two parallel instruction paths within a processor.

Reconsideration of claim 19 is respectfully requested.

Claim 20 depends from claim 19 and benefits from like argument. However, claim 20 has additional distinguishing features that patentably distinguish over Moore and Gupta. For example, claim 20 recites parallel instruction paths constructed and arranged to initially process first like instructions therethrough, the assessment logic monitoring the processing of the first like instructions to determine optimized heuristics for the instruction paths. Neither Moore nor Gupta disclose parallel instruction paths or assessment logic for monitoring the processing of instructions to determine optimized heuristics for the instruction paths. Reconsideration of claim 20 is respectfully requested.

11

Applicant believes no fees are due in connection with this Amendment and Response; however, if any fee is deemed necessary, the Commissioner is authorized to charge such fee to Deposit Account No. 08-2025.

Respectfully submitted,

By: _____Curt A. Vock_____

Curtis A. Vock, Reg. No. 38,356
LATHROP & GAGE L.C.
4845 Pearl East Circle, Suite 300
Boulder, CO 80301
Telephone: (720) 931-3011
Facsimile: (720) 931-3001

12